

CLAIMS

What is claimed is:

1. A method, comprising:

allocating a space to accommodate a mark bit and an allocation bit;

integrating the mark bit and the allocation bit into a mark/allocation bit using the

space; and

corresponding the mark/allocation bit with an object in a heap.
2. The method of claim 1, further comprising:

resetting the mark/allocation bit; and

switching the mark/allocation bit to the allocation bit to perform root set

enumeration.
3. The method of claim 2, further comprises performing the root set enumeration by

utilizing the mark/allocation bit as the allocation bit to conduct pointer

identification of the object in the heap.
4. The method of claim 3, further comprising:

resetting the mark/allocation bit; and

switching the mark/allocation bit to the mark bit to perform marking and scanning

of objects using the identified object.
5. The method of claim 4, further comprising:

marking the identified object as a root object;

scanning one or more objects associated with the root object by utilizing the

mark/allocation bit as the mark bit; and

- marking the scanned one or more objects associated with the root object.
6. The method of claim 5, further comprising:
- retaining the marked root object and the marked one or more objects associated with the root object; and
- regenerating allocation bits for the retained objects.
7. The method of claim 3, wherein the performing of the root set enumeration comprises lazy and selective root set enumeration.
8. The method of claim 7, wherein the lazy and selective root set enumeration comprises lazily and selectively identifying the root object in a segment of the heap and regenerating the allocation bits for the root object and other objects associated with the root objects residing in the segment of the heap.
9. A method, comprising:
- dividing a heap into a plurality of segments;
- using a mark/allocation bit as an allocation bit to locate a segment of the plurality of segments, the segment having a live object; and
- generating allocation bits for the located segment.
10. The method of claim 9, further comprising:
- determining whether the allocation bits have been generated for the located segment prior to generating the allocation bits; and
- generating the allocation bits for the located segment if the allocation bits have not been generated.
11. The method of claim 10, further comprising:
- determining whether an identification pointer points to a starting address or inside

- of the live object in the segment; and
- adding the live object to a root set if the identification pointer points to the starting address of the live object in the segment.
12. The method of claim 11, further comprising:
- calibrating the identification pointer to point to the starting address of the live object if the identification pointer points to the inside of the live object;
- verifying the identification pointer points to the starting address of the live object after calibration; and
- adding the live object to the root set if verified that the identification pointer points to the starting address of the live object in the segment.
13. The method of claim 9, wherein the allocation bit comprises a location bit to locate the segment of the plurality of segments in the heap.
14. A system, comprising:
- an allocation interface to allocate a space to accommodate a mark bit and an allocation bit;
- an integration module to integrate the mark bit and the allocation bit into a mark/allocation bit using the space; and
- a correspondence unit to correspond the mark/allocation bit with an object in a heap.
15. The system of claim 14, further comprising:
- a resetting routine to reset the mark/allocation bit;
- a switching module to switch the mark/allocation bit to the allocation bit to perform root set enumeration; and

a root set enumeration module to perform the root set enumeration by utilizing the mark/allocation bit as the allocation bit to conduct pointer identification of the object in the heap.

16. The system of claim 15, further comprising:

the resetting routine to reset the mark/allocation bit;

the switching module to switch the mark/allocation bit to the mark bit to perform marking and scanning using the identified object; and

a mark/scan unit to

mark the identified object as a root object,

scan one or more objects associated with the root object by utilizing the mark/allocation bit as the mark bit, and

mark the scanned one or more objects associated with the root object.

17. An apparatus, comprising

a processor; and

a storage medium coupled with the processor, the storage device having stored

thereon data representing sets of instructions, the sets of instructions

which, when executed by a processor, cause the processor to:

allocate a space to accommodate a mark bit and an allocation bit;

integrate the mark bit and the allocation bit into a mark/allocation bit using the space; and

correspond the mark/allocation bit with an object in a heap.

18. The apparatus of claim 17, wherein the processor is further to:

reset the mark/allocation bit;

switch the mark/allocation bit to the allocation bit to perform root set enumeration; and
perform the root set enumeration by utilizing the mark/allocation bit as the allocation bit to conduct pointer identification of the object in the heap.

19. The apparatus of claim 18, wherein the processor is further to:
reset the mark/allocation bit;
switch the mark/allocation bit to the mark bit to perform marking and scanning using the identified object;
mark the identified object as a root object;
scan one or more objects associated with the root object by utilizing the mark/allocation bit as the mark bit; and
mark the scanned one or more objects associated with the root object.
20. A machine-readable medium having stored thereon data representing sets of instructions which, when executed by a machine, cause the machine to:
allocate a space to accommodate a mark bit and an allocation bit;
integrate the mark bit and the allocation bit into a mark/allocation bit using the space; and
correspond the mark/allocation bit with an object in a heap.
21. The machine-readable medium of claim 20, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
reset the mark/allocation bit; and
switch the mark/allocation bit to the allocation bit to perform root set enumeration.

22. The machine-readable medium of claim 21, wherein the sets of instructions which, when executed by the machine, further cause the machine to perform the root set enumeration by utilizing the mark/allocation bit as the allocation bit to conduct pointer identification of the object in the heap.
23. The machine-readable medium of claim 22, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
reset the mark/allocation bit; and
switch the mark/allocation bit to the mark bit to perform marking and scanning using the identified object.
24. The machine-readable medium of claim 23, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
mark the identified object as a root object;
scan one or more objects associated with the root object by utilizing the mark/allocation bit as the mark bit; and
mark the scanned one or more objects associated with the root object.
25. The machine-readable medium of claim 24, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
retain the marked root object and the marked one or more objects associated with the root object; and
regenerate allocation bits for the retained objects.
26. The machine-readable medium of claim 22, wherein the performing of the root set enumeration comprises lazy and selective root set enumeration.

27. A machine-readable medium having stored thereon data representing sets of instructions, the sets of instructions which, when executed by a machine, cause the machine to:
- divide a heap into a plurality of segments;
- use a mark/allocation bit as an allocation bit to locate a segment of the plurality of segments, the segment having a live object; and
- generate allocation bits for the located segment.
28. The machine-readable medium of claim 27, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
- determine whether the allocation bits have been generated for the located segment prior to generating the allocation bits; and
- generate the allocation bits for the located segment if the allocation bits have not yet been generated.
29. The machine-readable medium of claim 28, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
- determine whether an identification pointer points to a starting address or inside of the live object in the segment; and
- add the live object to a root set if the identification pointer points to the starting address of the live object in the segment.
30. The machine-readable medium of claim 29, wherein the sets of instructions which, when executed by the machine, further cause the machine to:
- calibrate the identification pointer to point to the starting address of the live object if the identification pointer points to the inside of the live object;

verify the identification pointer points to the starting address of the live object

after calibration; and

add the live object to the root set if verified that the identification pointer points to

the starting address of the live object in the segment.